U-DADA: Unsupervised Deep Action Domain Adaptation

Arshad Jamal[1,2], Vinay P Namboodiri[2], Dipti Deodhare *, and KS Venkatesh[2]

[1] Centre for AI & Robotics, DRDO, Bangalore, India
[2] Indian Institute of Technology, Kanpur, India

Abstract. The problem of domain adaptation has been extensively studied for object classification task. However, this problem has not been as well studied for recognizing actions. While, object recognition is well understood, the diverse variety of videos in action recognition make the task of addressing domain shift to be more challenging. We address this problem by proposing a new novel adaptation technique that we term as unsupervised deep action domain adaptation (U-DADA). The main concept that we propose is that of explicitly modeling density based adaptation and using them while adapting domains for recognizing actions. We show that these techniques work well both for domain adaptation through adversarial learning to obtain invariant features or explicitly reducing the domain shift between distributions. The method is shown to work well using existing benchmark datasets such as UCF50, UCF101, HMDB51 and Olympic Sports. As a pioneering effort in the area of deep action adaptation, we are presenting several benchmark results and techniques that could serve as baselines to guide future research in this area.

Keywords: Action Recognition · Domain Adaptation · Transfer Learning.

1 Introduction

When a camera network is deployed for surveillance and security applications, the biggest challenge is to effectively use the visual recognition (object and human activity/event) algorithms trained on the dataset available to the developers. Often, these algorithms fail due to the problem, commonly known as *domain shift* between the data in the development and the real environment. While, domain shift has been widely studied in the context of object adaptation, there are hardly any effort to address this problem for action/event classification. In this paper, we investigate the domain shift in action space.

Deep Networks have been shown to bridge the gap between the source and target domains and learn transferable features. However, they cannot completely remove the gap between the two domains [11],[32]. To overcome this, several methods [14],[17],[18],[19],[32] have been proposed which incorporate additional

* The author is a former scientist from CAIR, DRDO, Bangalore, India

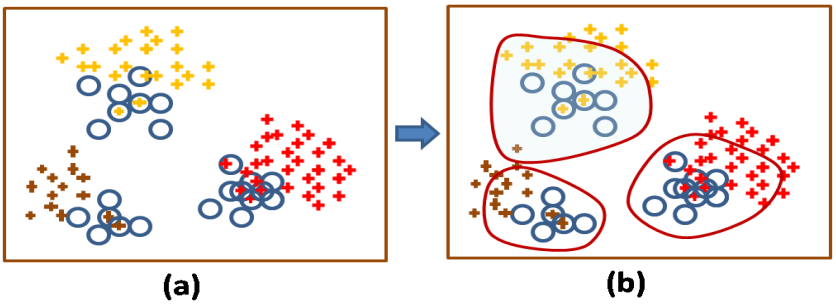


Fig. 1. Conceptual diagram of source sample selection to maximize positive transfer and minimize negative transfer. The samples of three classes from source domain are shown in three colors with '+' symbols. The unlabelled target domain samples are shown with 'o' symbol. Fig (b) shows the selected source samples. *Best viewed in color.*

layers into the deep network to align the source and target domain distributions and reduce domain discrepancy. In addition, there are other class of methods [10],[23],[31] which leverages the concept of Generative Adversarial Network [13] and formulate the problem as minimax game to make the source and target feature representations indistinguishable through adversarial learning. However, all these methods have been proposed mainly for object adaptation task and they have been shown to perform well for the standard object adaptation datasets.

In this paper, we investigate an equally important, but under-explored problem of action domain adaptation, which is even more challenging due to the diverse variety of videos. Human Action Recognition has been widely studied in the standard setting of supervised classification [4],[7],[24],[30],[35]. However, there are no efforts to evaluate these methods in multi-domain setting or embed domain adaptation architectures. We built an action domain adaptation architecture on top of the popular 3D-CNN [30] and evaluate it using three multi-domain datasets.

All the deep domain adaptation methods, mentioned above, blindly use the source domain dataset to align it with the target domain. Intuitively, it seems reasonable to expect that certain source data points, which are close to the target data points in the learned feature space would have positive effect on the adaptation process. However, there could be many other samples in the source domain that can spoil the alignment process. In this paper, we propose to address the problem by maximizing the positive transfer and minimizing the negative transfer from the source to the target domain. This is achieved by explicitly modeling density based adaptation and using them while adapting domains for recognizing actions. The idea has been illustrated in Fig 1 using a three class source and target dataset. We investigate two possibilities, one based on the density of the target points around each source point and another based on the density of the source points around the target points. These methods, we call as *Source Centred Target Density Modeling* (SCTDM) and *Target Centred Source Density*

Modeling (TCSDM) respectively. Empirically, we show that these techniques work well both for domain adaptation through adversarial learning to obtain invariant features or explicitly reducing the domain shift between distributions.

In summary, our main contributions are as follows:

1. Extend few popular object-centric deep domain adaptation methods for action adaptation and craft a new deep action domain adaptation method.
2. Propose a new guided learning framework for enhanced positive transfer and reduced negative transfer between source and target domain.
3. Extensive evaluation using several action datasets.

2 Related Work

Activity/Event analysis has been a widely studied area and a large number of papers have been published. However, the literature review, here, mainly focuses on various domain adaptation methods, which is a popular field in the area of transfer learning [21]. In a recent survey paper [2], domain adaptation and transfer learning techniques have been comprehensively discussed with a specific view on visual applications. It covers the historical shallow methods, homogeneous and heterogeneous domain adaptation methods and the deep domain adaptation methods that integrate the adaptation within the deep architecture.

Recently, deep domain adaptation methods [9],[17],[18],[19],[23] have shown significant performance gains over the prior shallow transfer learning methods. Many of these methods learn a feature representation in a latent space shared by the source and target domains. A popular approach among them is to minimize Maximum Mean Discrepancy (MMD) or its variant to effectively align the two distributions. Where, MMD is a non-parametric metric that measures the distribution divergence between the mean embedding of the two distributions in Reproducing Kernel Hilbert Space (RKHS). For example, in Deep Domain Confusion (DDC) method [32], the MMD is used in last fully connected layer along with classification loss to learn representations that are both domain invariant and discriminative. In Deep Adaptation Network (DAN) [17], Multi-Kernel MMD is used to improve the transferability of the features from source to target domain. In Residual Transfer Network (RTN) [18], the assumption of shared classifier between source and target domain is relaxed. It combines MK-MMD with an adaptive classifier to further improve the performance. The classifier is adapted by learning a residual function with reference to the target classifier. In Joint Adaptation Network (JAN) [19], Joint-MMD (JMMD) is used to align the joint distributions of multiple domain-specific layers across two domains. In another approach, simple linear transformation is used to align the second-order statistics of the source and target distributions. This approach, called as correlation alignment (COROL) [28] was further extended in [29] with Deep COROL in which a non-linear transformation is learned to the correlations of layer activation.

Other class of methods [10],[23],[31] for domain adaptation leverages the concept of Generative Adversarial Network (GAN) [13] and formulate the problem

as minimax game to learn a domain invariant feature representation. For example, in Domain Adversarial Neural Network (DANN) [10], gradient reversal layer is used for adversarial learning. In [31], a generic framework for adversarial adaptation is proposed in which the adversarial loss type with respect to the domain classifier and the weight sharing strategy can be chosen. In the adversarial learning methods, when the source and target features become completely indistinguishable, there are vanishing gradient problem. The Wasserstein Distance Guided Representation Learning (WDGRL) [23] method addresses the problem of vanishing distance.

All the domain adaptation methods discussed above are for image/object classification problem. The domain adaptation in videos has been highly underexplored. In fact, we could only find one subspace based method [15] on the video-to-video domain adaptation problem. There are few studies [3],[16],[26],[34],[36] on cross-view action recognition and a few on heterogeneous domain adaptation [5],[6],[33]. In that sense, to the best of our knowledge, this paper is one of the first few papers for the video-video domain adaptation.

3 Proposed Action Adaptation Approach

In this paper, the action domain adaptation networks are built on top of the feature embedding layers of a popular deep network architecture known as 3D-CNN [30]. This is either combined with the adversarial learning layer to obtain domain invariant features or distribution matching layer to explicitly reducing the domain shift between the distributions. The base convolutional layers of 3D-CNN learn mapping from the video input to a high-level feature space. The 3D-CNN network, when combined with the Gradient Reversal Layer (GRL), results in AGRL (Action GRL) and with Residual Transfer Network (RTN), it gives ARTN (Action RTN). We combine distribution matching and residual classifier learning with the GRL to create a unified adaptation framework named as unsupervised deep action domain adaptation (U-DADA), Further, we propose a density based adaptation approach, in which source samples are carefully selected to enhance positive transfer and reduce negative transfer between the two domains.

3.1 Problem Definition

Let's assume that the source domain consists of N_S labelled actions clips $\mathcal{D}_{\mathcal{S}} = \{\mathbf{x}_{\mathcal{S}}^i, y^i\}_{i=1}^{N_S}$, each having K-frames. Similarly, the target domain has N_T unlabelled action video clips, $\mathcal{D}_{\mathcal{T}} = \{\mathbf{x}_{\mathcal{T}}^i\}_{i=1}^{N_T}$, each having K-frames. The source and target domains are assumed to be sampled from different probability distributions $p_{\mathcal{S}}$ and $p_{\mathcal{T}}$ respectively, and $p_{\mathcal{S}} \neq p_{\mathcal{T}}$. The goal of this paper is to design a deep action domain adaptation network that learns feature embedding $\mathbf{f} = G_f(x)$ and transfer classifiers $y = G_y(f)$, such that the expected target risk $Pr_{(x,y)\sim p_{\mathcal{T}}}[G_y(G_f(x)) \neq y]$ can be bounded by leveraging the source domain labeled data.

3.2 Preliminaries

Maximum Mean Discrepancy: Let the source and target data be sampled from probability distributions $p_{\mathcal{S}}$ and $p_{\mathcal{T}}$ respectively. Maximum Mean Discrepancy (MMD) [14] is a kernel two-sample test which rejects or accepts the null hypothesis $p_{\mathcal{S}} = p_{\mathcal{T}}$ based on the observed samples. Formally, MMD is defined as,

$$D_{\mathcal{H}}(p_{\mathcal{S}}, p_{\mathcal{T}}) = \sup_{h \in \mathcal{H}} \left[\mathbb{E}_{\mathbf{X}_{\mathcal{S}}}[h(\mathbf{X}_{\mathcal{S}}] - \mathbb{E}_{\mathbf{X}_{\mathcal{T}}}[h(\mathbf{X}_{\mathcal{T}}] \right], \tag{1}$$

where $\mathcal{H}$ is a class of function lying in RKHS (Reproducing Kernel Hilbert Space), which can distinguish any two distribution. In this case, MMD is the distance between their mean embedding: $D_{\mathcal{H}}(p_{\mathcal{S}}, p_{\mathcal{T}}) = \|\mu_{\mathbf{X}_{\mathcal{S}}}(p_{\mathcal{S}}) - \mu_{\mathbf{X}_{\mathcal{T}}}(p_{\mathcal{T}})\|^2_{\mathcal{H}}$. Theoretically, it has been shown [14] that $p_{\mathcal{S}} = p_{\mathcal{T}}$ if and only if $D_{\mathcal{H}}(p_{\mathcal{S}}, p_{\mathcal{T}}) = 0$. In practice, the MMD is estimated using the following equation:

$$\begin{aligned} \hat{D}_{\mathcal{H}}(p_{\mathcal{S}}, p_{\mathcal{T}}) = \frac{1}{N_{\mathcal{S}}^2} \sum_{i=1}^{N_{\mathcal{S}}} \sum_{j=1}^{N_{\mathcal{S}}} k(\mathbf{x}_{\mathcal{S}}^i, \mathbf{x}_{\mathcal{S}}^j) + \frac{1}{N_{\mathcal{T}}^2} \sum_{i=1}^{N_{\mathcal{T}}} \sum_{j=1}^{N_{\mathcal{T}}} k(\mathbf{x}_{\mathcal{T}}^i, \mathbf{x}_{\mathcal{T}}^j) \\ - \frac{2}{N_{\mathcal{S}} N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{S}}} \sum_{j=1}^{N_{\mathcal{T}}} k(\mathbf{x}_{\mathcal{S}}^i, \mathbf{x}_{\mathcal{T}}^j), \end{aligned} \tag{2}$$

where, $\hat{D}_{\mathcal{H}}(p_{\mathcal{S}}, p_{\mathcal{T}})$ is the unbiased estimate of $D_{\mathcal{H}}(p_{\mathcal{S}}, p_{\mathcal{T}})$. The characteristic kernel $k(x^i, x^j) = e^{\|vec(x^i) - vec(x^j)\|^2 / b}$ is the Gaussian kernel function defined on the vectorization of tensors x^i and x^j with bandwidth parameter b.

3.3 Deep Action Domain Adaptation (DADA)

In this paper, we craft a new action domain adaptation network, which is built on the popular deep network architecture known as 3D-CNN [30]. The proposed network, incorporates several popular choices from object adaptation space. This network, named as Deep Action Domain Adaptation (DADA), combines the adversarial learning and distribution matching ideas from the image adaptation literature. Our network architecture, shown in Fig 2, includes seven layers of 3D-CNN for feature mapping $G_f(.;\theta_f)$, 3-5 layers of residual network for classifier adaptation $G_y(.;\theta_y)$, 1-3 layers of MK-MMD for feature distribution matching $D_{\mathcal{L}}(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})$, one layer for entropy of class-conditional distribution of target data $G_E(.;\theta_E)$ and three adversarial layers for domain alignment $G_d(.;\theta_d)$. In the network, the feature mapping layers share weight between source and target domains. Here, an adversarial game is played between a domain discriminator $G_d(.;\theta_d)$, which is trained to distinguish the source and target domain samples, and the feature extractor $G_f(.;\theta_f)$, which is fine-tuned simultaneously to confuse the domain discriminator. Similarly, the other layers are fine-tuned to minimize the losses.

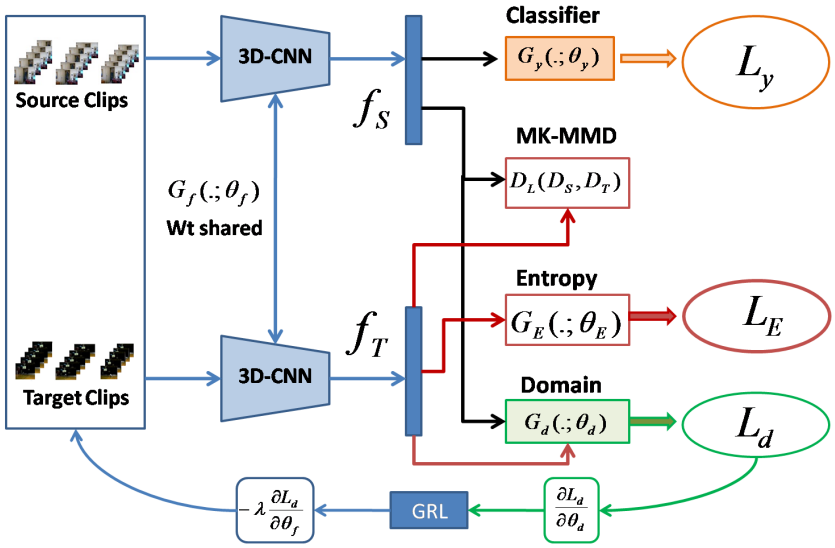


Fig. 2. Architecture of the proposed Deep Action Domain Adaptation (DADA) network. $G_f(.;\theta_f)$ is the feature mapping function, $G_y(.;\theta_y)$ is the class discriminator function, $G_E(.;\theta_E)$ is the entropy function and $G_d(.;\theta_d)$ is the domain discriminator function. L_y, L_E and L_d are the corresponding loss functions. $D_{\mathcal{L}}(\mathcal{D}_S, \mathcal{D}_T)$ is the distribution matching function. GRL is gradient reversal layer. Other back-propagation layers have been omitted for simplicity. *Best viewed in color.*

In the adversarial training, the parameters θ_f are learned by maximizing the domain discriminator loss L_d and the parameters θ_d are learned by minimizing the domain loss. In addition, the label prediction loss L_y, the MK-MMD loss $D(\mathcal{D}_S, \mathcal{D}_T)$ and target data entropy L_E are also minimized. The overall loss function for the DADA is:

$$\begin{aligned} L(\theta_f, \theta_y, \theta_E, \theta_d) = & \frac{1}{N_s} \sum_{x_i \in \mathcal{D}_S} L_y(G_y(G_f(x_i)), y_i) \\ & + \gamma D_{\mathcal{L}}(\mathcal{D}_S, \mathcal{D}_T) + \frac{\mu}{N_T} \sum_{x_i \in \mathcal{D}_T} L_E(G_E(G_f(x_i))) \\ & - \frac{\lambda}{N_S + N_T} \sum_{x_i \in \mathcal{D}_S \cup \mathcal{D}_T} L_d(G_d(G_f(x_i)), d_i), \end{aligned} \tag{3}$$

where γ, μ and λ are the trade-off parameters in the objective function (3) that shape the features during learning. L_y is the cross-entropy loss for label prediction, L_E is the entropy function of class-conditional distribution of the target features [18] and L_d is the domain classification loss [9]. At the end of the training, the parameters $\hat{\theta}_f$, $\hat{\theta}_y$, $\hat{\theta}_d$ will give the saddle point of

the loss function (3): $(\hat{\theta_f}, \hat{\theta_y}, \hat{\theta_E}) = \text{argmin}_{\theta_f, \theta_E, \theta_y} L(\theta_f, \theta_y, \theta_E, \theta_d)$ and $(\hat{\theta_d}) = \text{argmax}_{\theta_d} L(\theta_f, \theta_y, \theta_E, \theta_d)$.

3.4 Density Based Adaptation

The existing deep object domain adaptation methods blindly use all the source domain samples without worrying about the transfer capabilities of the individual samples. Intuitively, it seems reasonable that certain source data samples, which are close to the target data samples in the learned feature space would have positive transfer and other samples which are far off in the feature space would have negative transfer. This makes the training unstable and compromises the domain transfer capability of various methods. In this paper, we address the problem by explicitly modeling density based adaptation, which enhances the positive transfer and reduces the negative transfer from the source to the target domain. This is achieved by an informed selection of a subset of source domain points based on their *closeness* with the target domain. The concept of source selection has been illustrated in Fig 1 using three class domain adaptation problem. We propose two approaches, one based on the density of the target samples around each source sample and another based on the density of the source samples around the target samples. These methods, called as *Source Centred Target Density Modeling* (SCTDM) and *Target Centred Source Density Modeling* (TCSDM) are discussed below.

Source Centred Target Density Modeling (SCTDM): Let us define the number of target points around each source point as:

$$n_T(\mathbf{x_S}) = |\mathbf{x_T}| Sim(\mathbf{x_S}, \mathbf{x_T}) \geq \epsilon| \tag{4}$$

where, ϵ is the mean similarity between each source point and all the target points. The similarity measure, $Sim(\mathbf{x_S}, \mathbf{x_T}) = \mathbf{x_S} G \mathbf{x'_T}$, where G is a similarity kernel. There are several options available for G, which can be used as similarity kernel between source and target domain (e.g. Radial basis kernel). However, in this work, G is set to identity matrix. Further, we define the average target density for each class of the source data as,

$$\overline{n}_{\mathcal{T}}^{(c)} = \frac{1}{N_S^c} \sum (n_{\mathcal{T}}(\mathbf{x_S}) | \mathbf{y_S} = c) \tag{5}$$

where N_S^c is the number of source data points in class-c and $\mathbf{y}_S$ is the class label.

Given a source and target dataset $\mathbf{X}_S \in \mathbb{R}^{N_S \times D}$ and $\mathbf{X}_T \in \mathbb{R}^{N_T \times D}$, consisting of N_S and N_T D-dimensional feature vectors computed using the fine-tuned model. We compute the similarity $Sim(\mathbf{x_S}, \mathbf{x_T})$ between the source and target domain and the average target density $\overline{n}_{\mathcal{T}}^{(c)}$ for each class in the source domain and then select a balanced set of source samples having target density more than the $\overline{n}_{\mathcal{T}}^{(c)}$ i.e. $\mathbf{X'_S} = \left\{ \mathbf{x_S} : n_{\mathcal{T}}(\mathbf{x_S}) > \overline{n}_{\mathcal{T}}^{(c)}, |\mathbf{X'_S}| < \alpha |\mathbf{X_S}| \right\}$. Here, α is a hyper parameter defining the fraction of the source data, the upper limit of which is fixed at 90% in all the experiments.

Target Centred Source Density Modeling (TCSDM): In the second approach, as illustrated in Fig 1, we perform clustering of the target data, centred around which, we find a balanced set of samples from the source domain. If the number of classes in labelled source data is C, the target data is clustered into C-clusters. For each cluster $c \in C$, its radius R_c is found and also the distance between all the source points with C clusters are computed (a distance matrix ($N_S \times C$). The distances are sorted to find the closest cluster for each source point. Now, for each class, the source points are examined in the order of their increasing distance from the cluster centre. The points are selected if they are within a distance of $R_c/2$ (empirically chosen based on the analysis of distances for the dataset) from the cluster centre and the total count of the particular source class is less than a predefined fraction. The percentage of source data to be selected is a hyper-parameter, whose upper limit is fixed at 90% in all the experiments.

4 Experiments

In this paper, our deep action domain adaptation method has been compared with two other deep adaptation methods, which are obtained by extending two object domain adaptation methods. They are adversarial learning based Gradient Reversal Layer (**AGRL** - Action GRL) method and feature distribution alignment method (**ARTN** - Action Residual Transfer Network). In addition, three baselines methods i.e. **3D-CNN**, **AGFK** (action variant of Geodesic Flow Kernel [12] method) and **ASA** (action variant of Subspace Alignment [8] method) have also been used in the experiments. Here, the first baseline provides the No Adaptation results. The details of the experimental setup including action datasets are discussed in the following subsections.

4.1 Setup

The action DA experiments require multiple distinct action datasets having the same action categories. Unfortunately, there are hardly any benchmark action datasets available for this experiment. We specifically created three multi-domain datasets, as described below, and evaluated the proposed approaches with them. Specifically, for the deep action adaptation methods, a larger eighteen class multi-domain dataset has been created from UCF101 dataset as one domain and a combination of Olympic Sports and HMDB51 datasets as the other domain. In all the cases, publicly available Sport 1M 3D-CNN model is fine-tuned using the source domain data, which is then used in the adaptation problems. The dataset details are as follows.

KTH, MSR Action II and SonyCam Datasets: Our first dataset collection, referred to as **KMS**, is a combination of three datasets, consisting of two benchmark datasets i.e. KTH [1] and MSR Action II [1] (denoted by **K** and **M** respectively) along with a six class dataset, referred to as SonyCam (denoted by

S) captured using a **hand-held** Sony camera. In KTH and SonyCam datasets, there are six classes, namely, *Boxing, Handclapping, Handwaving, Jogging, Running* and *Walking*. MSR Action II dataset contains only the first three classes from the KTH dataset. For the **KMS** dataset collection, there are four adaptation problems (K $\rightarrow$ M, K $\rightarrow$ S, M $\rightarrow$ S and M $\rightarrow$ K). The SonyCam dataset is only used as target domain owing to its small size (180 clips across 6 action classes). In case of KTH dataset, we use training data partition of 1530 clips spread almost equally across six classes for source domains and testing data partition of 760 clips for target domain. In the MSR dataset, there are 202 clips for three classes (Boxing-80, Handclapping-51 and Handwaving-71).

UCF50 and Olympic Sports Datasets: The second dataset collection comprises a subset of six common classes from UCF50 [22] (denoted by **U**) and Olympic Sports [20] (denoted by **O**). The classes are *Basketball, Clean and Jerk, Diving, Pole Vault, Tennis* and *Discus Throw*. For UCF50 dataset, we use 70%-30% train-test split suggested in [27], which results into $432 - 168$ train/test action videos. Each of them are then segmented into 16-frames clips for training and testing. Similarly, for Olympic Sports dataset, the number of unsegmented videos in training and testing set are 260 and 55 respectively. In this case, **U** $\rightarrow$ **O** and **O** $\rightarrow$ **U** are the two adaptation problems being solved.

Olympic Sports, HMDB51 and UCF101 Datasets: In the third series of experiments, we combined the Olympic Sports and HMDB51 datasets (denoted by **OH**) to construct a much larger multi-domain dataset and used all the eighteen common classes between OH and UCF101 dataset. The eighteen common classes are *Basketball, Biking, Bowling, Clean and Jerk, Diving, Fencing, Golf Swing, Hammer Throw, High Jump, Horse Riding, Javelin Throw, Long Jump, Pole Vault, Pull-ups, Push-ups, Shot-put, Tennis Swing* and *Throw Discus*. The name of the classes varies slightly across the three datasets. For UCF101, the splits suggested in [25] has been used, which results in 2411 segmented videos of 32-frames distributed between train and test set of 1712 and 699 video clips. Similarly, for the **OH** combination, 70%-30% split has been used for training and testing, which results in 958 and 303 video clips for the two. In this case, **OH** $\rightarrow$ **UCF** and **UCF** $\rightarrow$ **OH** are the two DA problems.

4.2 Implementation Details

For the feature embedding, we used the 3D-CNN architecture [30]. All the subspace based domain adaptation experiments have been conducted using the 4096-dimensional *fc7* features computed for 16-frame clips, obtained by segmenting the action videos. The source and target domain points on the subspace are obtained by separately stacking all the features corresponding to the two domains and then computing the PCA of the resulting matrices.

In the case of deep action adaptation, we combine the feature mapping layers of the 3D-CNN with the layers of gradient reversal layer and distribution

alignment layer to correspondingly obtain the **AGRL** and **ARTN** methods. Similarly, the proposed **DADA** network is built on top of the 3D-CNN model by incorporating all the components mentioned in Section 3.3. The inputs to the network are the mini-batch of the video clips. The classification is done using the softmax layer. We fine-tune all convolutional and pooling layers and train the classifier layer via back propagation. Since the classifier is trained from scratch, we set its learning rate to be 10 times that of the lower layers. We employ the mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and the learning rate strategy implemented in RevGrad [9].

The proposed density based adaptation approach starts with computation of the feature vectors using the fine-tuned network. These features are then used to find a subset of source domain data samples using either of the two methods discussed in Section 3.4 and 3.4. Once, the source samples are selected, the end-to-end training is performed using the respective adaptation methods discussed above.

4.3 Results and Discussions

In this section, we first present the results of all the deep action adaptation methods for the three dataset collections. Then the effect of our proposed source subset selection approach is analyzed for these methods.

Domain Adaptation in Action Spaces: We have evaluated our action domain adaptation approaches for **UO**, **KMS** and **OH-UCF101** dataset collections. In majority of the cases, improvements have been observed over all the baselines. The two subspace based domain adaptation methods (i.e. AGFK and ASA) have been found to be generally better than 3D-CNN and the deep domain adaptation approaches substantially outperforms all the three baselines. In Table 1, results for the **KMS** and **UO** datasets have been given. It can be seen that the subspace based adaptation methods are better than the 3D-CNN No-Adaptation baseline, which was significantly improved by the **AGRL**, **ARTN** and **DADA** methods. In four out of six adaptation problems, the proposed DADA approach gives best results. In this table, all the three deep adaptation methods use TCSDM approach for source sample selection. In Table 2, the adaptation results for the **OH-UCF101** dataset has been given. There are substantial improvement over the three baseline, shown in the first three columns. The proposed DADA approach outperforms the other two deep adaptation methods for both the adaptation problems. Here, the three deep adaptation methods use SCTDM approach for source sample selection.

The results obtained for the action domain adaptation confirms the earlier findings of the object domain adaptation methods in [9],[18]. The domain adaptation module, when integrated with the 3D-CNN, improves the domain adaptation performance. Moreover, the proposed adaptation network, incorporating the adversarial learning and multi-kernel two-sample matching, further improves the adaptation performance.

Table 1. Action Domain Adaptation results for the **KMS** and **UO** datasets. 4096-dimensional *fc7* features are used for subspace based adaptation methods AGFK and ASA. All the deep methods (last three rows) use TCSDM approach for density based adaptation. The best results are shown in bold

Methods	K → S	K → M	M → S	M → K	U → O	O → U
3D-CNN [30]	61.11	49.8	70.22	71.89	82.13	83.16
AGFK	63.71	61.16	73.27	72.9	84.04	86.21
ASA	64.71	62.13	76.7	74.5	84.10	85.67
AGRL	70.44	73.2	77.33	86.85	88.65	91.6
ARTN	72.55	73.6	77.1	**97.38**	87.45	**92.58**
DADA	**73.11**	**76.6**	**77.78**	96.66	**93.01**	91.3

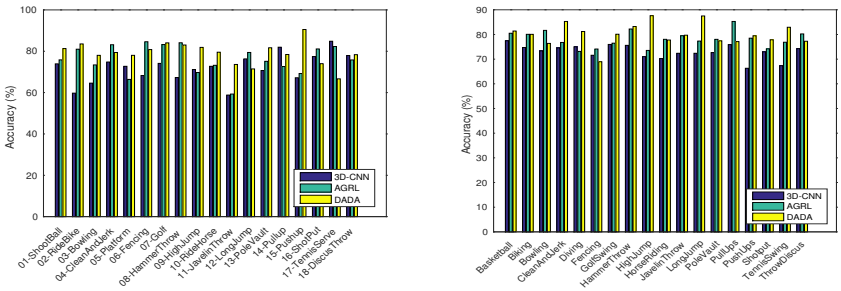
Table 2. Action Domain Adaptation results for the **OH-UCF101** datasets. 4096-dimensional *fc7* features are used for subspace based adaptation methods AGFK and ASA. All the deep methods (last three columns) use SCTDM approach for density based adaptation. The best results are shown in bold

	3D-CNN [30]	AGFK	ASA	AGRL	ARTN	DADA
UCF → OH	72.24	72.31	70.57	76.37	**79.13**	**79.13**
OH → UCF	72.92	75.45	75.02	78.1	78.45	**80.17**

Analysis of Deep Action Domain Adaptation Learning: The performance of the three adaptation methods (3D-CNN, AGRL and DADA) across eighteen classes of OH-UCF101 dataset are shown in Fig 3. For $OH \rightarrow UCF$ adaptation, the proposed DADA architecture outperforms the other methods for 12-classes and the AGRL method is best for other 6-classes. Similarly, for $UCF \rightarrow OH$ adaptation, our approach outperforms the other methods for 11-classes and the AGRL method is best for 5-classes and in other two classes 3D-CNN is best. In other experiments also, similar results were obtained.

The source sample selection methods (i.e. TCSDM and SCTDM) have been evaluated for the proposed DADA network using KMS and OH-UCF101 datasets. The results, as shown in Table 3, clearly demonstrates the positive effect of the informed source selection. In all the six adaptation problems, across the two datasets, the proposed density based adaptation improves the results over the full data training. Similarly, the improvements were also obtained for the other deep adaptation methods, the results for which are shown in Fig 5.

The source samples not selected by the two methods were visually scrutinized to understand the reasons of their non-selection. There were three main observations: (i) the video clips were visually far away from the action class; (ii) the video clips had no action performed; (iii) the action are not visible due to clutter or only partially visible due to occlusion. Few example video clips, illustrating these observations, are shown in Fig 4.



(a) (b)

Fig. 3. Adaptation performance across eighteen classes of UCF101-OH datasets for **3D-CNN**, **AGRL**, **DADA** methods (a) $UCF101 \rightarrow OH$ (b) $OH \rightarrow UCF101$.

Table 3. Comparative analysis of the source selection methods for the **OH-UCF101** and **KMS** datasets. The best results are shown in bold

Methods	UCF → OH	OH → UCF	K → S	K → M	M → S	M → K
Full Source Data	**79.13**	78.45	71.33	73.2	77.3	89.1
SrcSel-TCSDM	78.96	79.78	73.11	**76.6**	**77.78**	**96.66**
SrcSel-SCTDM	**79.13**	**80.07**	**74.08**	74.91	76.89	87.76

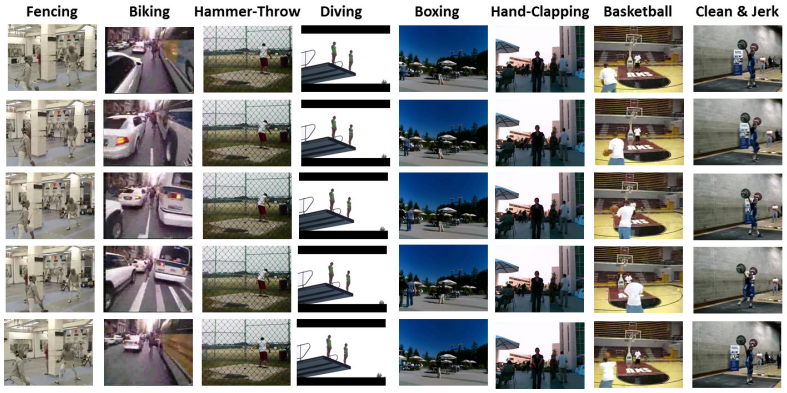
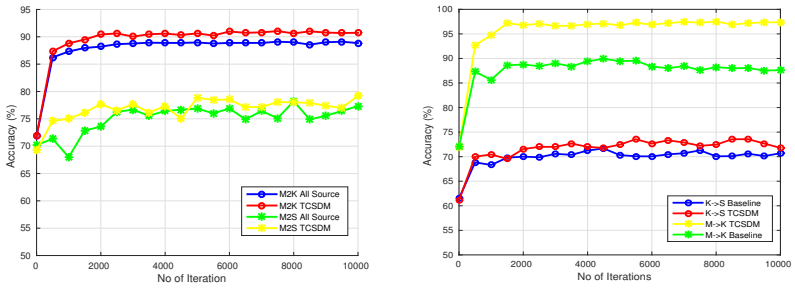


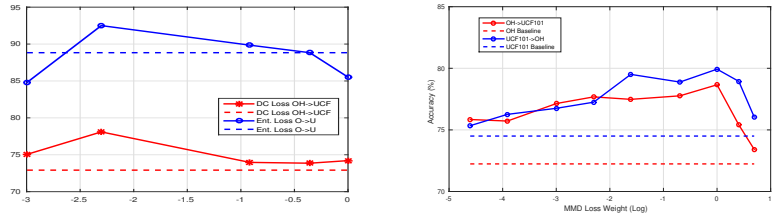
Fig. 4. Source video clips, not selected by the TCSDM method, for few classes in the KMS and OH-UCF101 dataset. Each column has 5 sampled frames of a video clip of the mentioned classes.

Effect of the Source Sample Selection on Training: The density based adaptation, discussed in Section 3.4 is evaluated using the KMS dataset for both **AGRL** and **ARTN** methods. The results are shown in Fig 5. In each figure, two pair of graphs are shown, one for training using all the source data and the other for the proposed density based adaptation using a subset of source samples selected using the methods discussed in Section 3.4. The effect of enhanced positive transfer and reduced negative transfer due to the informed selection of source subset is visible in all the graphs. Specifically, for $M \rightarrow K$ adaptation problem, the **ARTN** method gave significant jump for density based adaptation.



(a) (b)

Fig. 5. Improvement due to guided learning: Accuracy vs. Iterations; (a) $K \rightarrow S$ and $M \rightarrow K$ for **AGRL** method (b) $M \rightarrow K$ and $M \rightarrow S$ for **ARTN** method.



(a) (b)

Fig. 6. Trade-off Parameter Selection: (a) Accuracy vs. Entropy Loss plot for $O \rightarrow U$ and Accuracy vs. DC Loss plot for $OH \rightarrow UCF101$ (b) Accuracy vs. MMD Loss plat for $OH \rightarrow UCF101$ and $UCF101 \rightarrow OH$ for **ARTN** method. All baseline results are shown as dotted line. *Best viewed in color*

Hyper-parameter Selection: In the experiments, few hyper-parameters have been chosen based on either the recommendations given in the respective papers or the specific experiments done in this paper. For example, the trade-off

parameters μ, γ and λ in the optimization function (3) have been empirically selected based on the domain adaptation experiments. In order to minimize the search space of these individual parameters, greedy approach has been used. For example, the entropy loss parameter μ and MK-MMD parameter γ is selected by running the experiments for ARTN method, which is then used in Eq. (3) to selected λ.

The search space for λ and μ are $\{0.05, 0.1, 0.4, 0.7, 1.0\}$. The variation in the classification accuracy for these two parameters is shown in Fig 6(a). The domain confusion parameter λ was found to be 0.1 and 0.4 for different adaptation problems and the entropy loss parameter was found to be 0.1. Similarly, the MK-MMD parameter γ is selected by running the experiments for the values of $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 1.5, 2.0\}$. The variations in classification accuracy with MMD Loss weight for $OH \rightarrow UCF101$ and $UCF101 \rightarrow OH$ are shown in Fig 6(a). The maximum accuracy was obtained for MMD Loss 1.0. In these experiments, as shown in Fig 6, a bell-shaped curve is obtained as the accuracy first increases and then decreases with the variation in trade-off parameters.

The learning rate was another hyper-parameter, which was empirically selected. Experiments were done for learning rate between $0.01 - 0.0001$ using the **UO** dataset. The learning rate was reduced by a factor of $1/\sqrt{10}$. In this experiments, the maximum accuracy was obtained for 0.0001. For all other datasets, we have used the same learning rate.

5 Conclusions and future work

In this paper, we formulated the problem of domain adaptation for human action recognition and extended two popular approaches of object adaptation for deep action adaptation. We crafted a new deep domain adaptation network for action space. The methods have been comprehensively evaluated using three multi-domain dataset collections, one of which is a large eighteen class collection. We compare these methods with three baselines and show that our deep action domain adaptation method perform better then the baselines. Further, we proposed a new density based adaptation method to enhance the positive transfer and reduce the negative transfer between the source and target domains. Consistent and significant performance improvements have been obtained across various experiments. In this paper, several benchmark results and techniques have been proposed that could serve as baselines to guide future research in this area. In future, we would like to study the concept of continuous domain adaptation on the streaming action videos. In addition, we would like to study other deep learning frameworks for action domain adaptation.

Acknowledgment

The authors would like to thank the Director, Centre for AI & Robotics, Bangalore, India for supporting the research.

References

1. KTH and MSR action II dataset http://www.cs.utexas.edu/ chaoyeh/web_action-_data/dataset_list.html
2. Csurka, G.: A comprehensive survey on domain adaptation for visual applications. In: Domain Adaptation in Computer Vision Applications. pp. 1–35 (2017)
3. Davar, N.F., deCampos, T.E., Windridge, D., Kittler, J., Christmas, W.: Domain adaptation in the context of sport video action recognition. In: Domain Adaptation Workshop, in conjunction with NIPS (2011)
4. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
5. Duan, L., Xu, D., Tsang, I.W.: Domain adaptation from multiple sources: A domain-dependent regularization approach. IEEE Transactions on Neural Networks and Learning Systems **23**(3), 504–518 (March 2012)
6. Duan, L., Xu, D., Tsang, I.W., Luo, J.: Visual event recognition in videos by learning from web data. PAMI **34**(9), 1667–1680 (September 2012)
7. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR 2016. pp. 1933–1941
8. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV 2013. pp. 2960–2967
9. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32th International Conference on Machine Learning ICML 2015, Lille, France, 6-11 July 2015. pp. 1180–1189 (2015)
10. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. Journal of Machine Learning Research **17**, 59:1–59:35 (2016)
11. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. ICML'11 (2011)
12. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR 2012. pp. 2066–2073 (2012)
13. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial networks. CoRR **abs/1406.2661** (2014)
14. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. J. Mach. Learn. Res. **13**(1), 723–773 (Mar 2012)
15. Jamal, A., Deodhare, D., Namboodiri, V., Venkatesh, K.S.: Eclectic domain mixing for effective adaptation in action spaces. Multimedia Tools and Applications **77**(22), 29949–29969 (Nov 2018). https://doi.org/10.1007/s11042-018-6179-y
16. Li, R.: Discriminative virtual views for cross-view action recognition. In: CVPR 2012. pp. 2855–2862 (2012)
17. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32th International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. pp. 97–105 (2015)
18. Long, M., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. CoRR **abs/1602.04433** (2016), http://arxiv.org/abs/1602.04433
19. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th International Conference on Machine

Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 2208–2217 (2017)
20. Niebles, J.C., Chen, C.W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. pp. 392–405. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
21. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Knowl. and Data Eng. **22**(10), 1345–1359 (Oct 2010)
22. Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. Mach. Vision Appl. **24**(5), 971–981 (Jul 2013)
23. Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: AAAI. AAAI Press (2018)
24. Shou, Z., Wang, D., Chang, S.: Temporal action localization in untrimmed videos via multi-stage cnns. In: CVPR. pp. 1049–1058. IEEE Computer Society (2016)
25. Soomro, K., Zamir, A.R., Shah, M., Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. CoRR (2012)
26. Sui, W., Wu, X., Feng, Y., Jia, Y.: Heterogeneous discriminant analysis for cross-view action recognition. Neurocomput. **191**(C), 286–295 (May 2016), https://doi.org/10.1016/j.neucom.2016.01.051
27. Sultani, W., Saleemi, I.: Human action recognition across datasets by foreground-weighted histogram decomposition. In: CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 764–771 (2014)
28. Sun, B., Feng, J., Saenko, K.: Correlation alignment for unsupervised domain adaptation. CoRR **abs/1612.01939** (2016)
29. Sun, B., Saenko, K.: Deep CORAL: correlation alignment for deep domain adaptation. In: Hua, G., Jégou, H. (eds.) Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9915, pp. 443–450
30. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV 2015, Santiago, Chile, December 7-13, 2015. pp. 4489–4497 (2015)
31. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 2962–2971 (2017)
32. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. CoRR **abs/1412.3474** (2014), http://arxiv.org/abs/1412.3474
33. Wang, H., Wu, X., Jia, Y.: Video annotation via image groups from the web. IEEE Transactions on Multimedia **16**(5) (August 2014)
34. Wu, X., Wang, H., Liu, C., Jia, Y.: Cross-view action recognition over heterogeneous feature spaces. In: IEEE International Conference on Computer Vision. pp. 609–616 (Dec 2013). https://doi.org/10.1109/ICCV.2013.81
35. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5794–5803 (2018)
36. Zhang, Z., Wang, C., Xiao, B., Zhou, W., Liu, S., Shi, C.: Cross-view action recognition via a continuous virtual path. In: CVPR 2013, Portland, OR, USA, June 23-28, 2013. pp. 2690–2697 (2013)